



# *CE Workgroup*

## **Status of Embedded Linux**

**May 2023**

Tim Bird

Principal Software Engineer, Sony Electronics



CE Workgroup

# Nature of this talk...

- I periodically take a look at the status of embedded Linux
- This talk will be a weird hybrid of:
  - Talk on kernel technologies and industry news
  - Thoughts on the overall status of major technology areas and the embedded Linux ecosystem
- Hope to accomplish 2 things:
  - Let people know what's going on
  - Open a discussion on 'what's next' (and what needs more work)



CE Workgroup

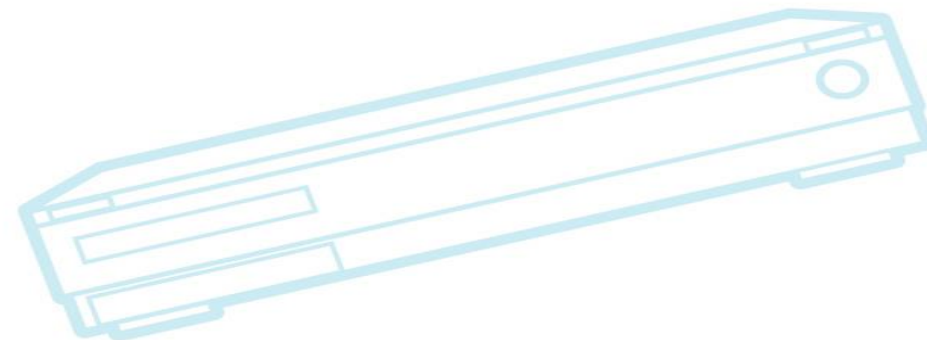
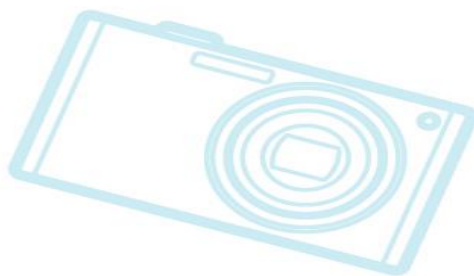
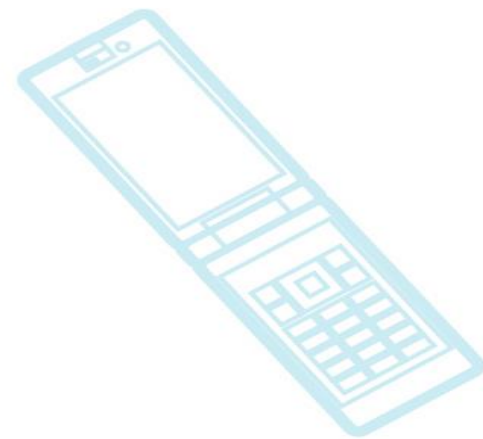
# Embedded Linux past vs. present



CE Workgroup

# Overall agenda

- History (past)
- Status (present)
- Are we done yet?







CE Workgroup

# A long time ago...

- In 1998, I started working on embedded Linux at Lineo
- In 2003, I was hired by Sony to help start the “Consumer Electronics Linux Forum” (CELF)
  - In 2005, CELF started the Embedded Linux Technical Conference
- In 2010, CELF became part of the Linux Foundation that became the “Core Embedded Linux Project” (CELP)



# In the early days...

- In the '90s we spent lots of effort convincing companies that Linux was suitable for embedded devices
  - Competition back then was: VRTX, VxWorks, pSOS, Nucleus, µltron (ie Commercial RTOSes)
- Architecture support: Intel, ARM, MIPS
  - But, no major chip vendor wrote arch support for Linux
  - It was all community developers, or nascent embedded Linux companies
- Embedded distros: Roll your own, or MontaVista, Lineo, TimeSys



CE Workgroup

# Focus areas back then

- CELF spent time analyzing and trying to address deficiencies in Linux, to make it suitable for embedded
- In 2003, CELF identified 5 key areas that needed work:
  - System size
  - Boot time
  - Power management
  - Realtime
  - Security
- CELF funded the contribution of several features in the Linux kernel in these and other areas (like filesystems)



CE Workgroup

# Are we done yet?

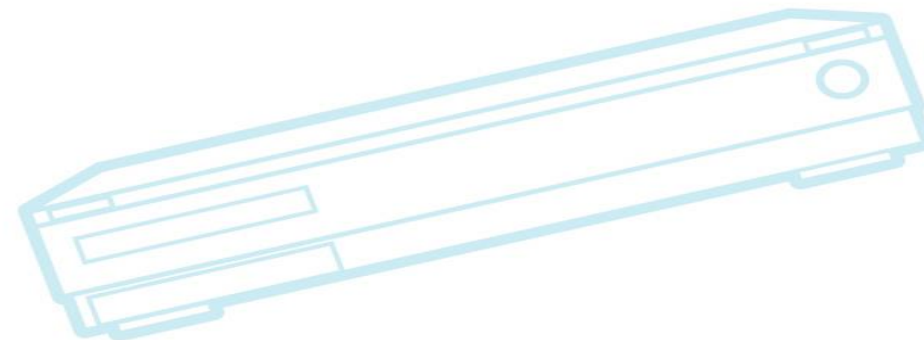
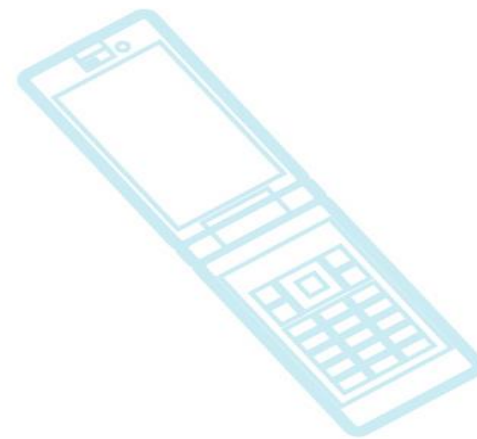
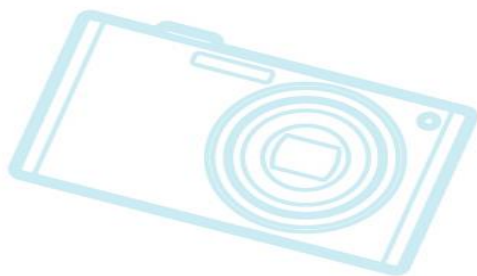
- I am still haunted by a phrase said in jest (I think) at ELC 2008 by Andrew Morton
  - “Are we done yet?”
- BTW: Andrews’ talk that year is available, and it still applicable for companies just getting into embedded Linux
  - See <https://bootlin.com/pub/video/2008/elc/elc2008-andrew-morton-keynote.ogg>





CE Workgroup

# Present: Linux Today

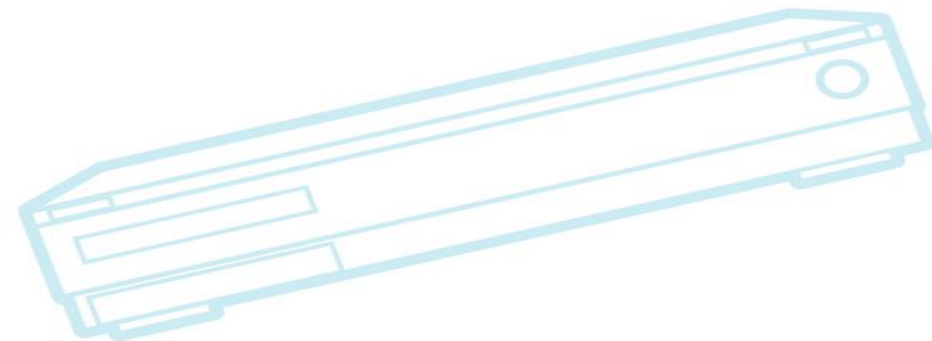
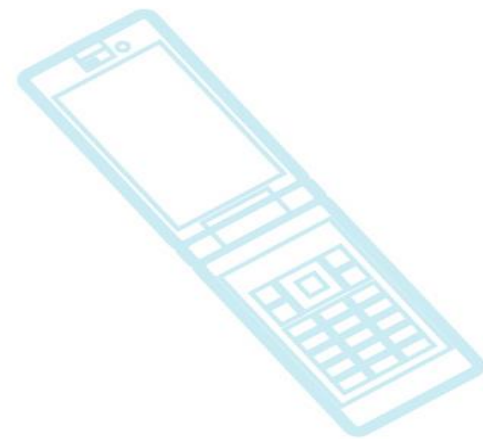
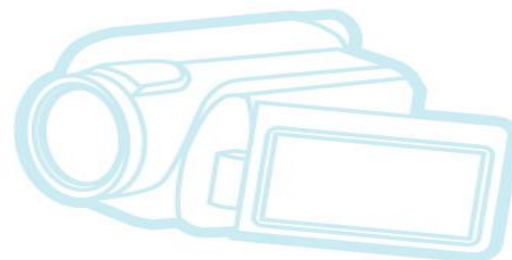




CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Industry News  
Community  
Scorecards  
Conclusions





CE Workgroup

# Outline

## Linux Kernel

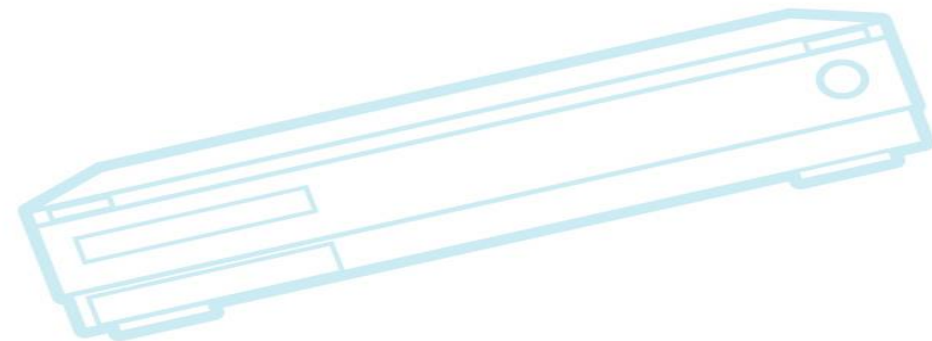
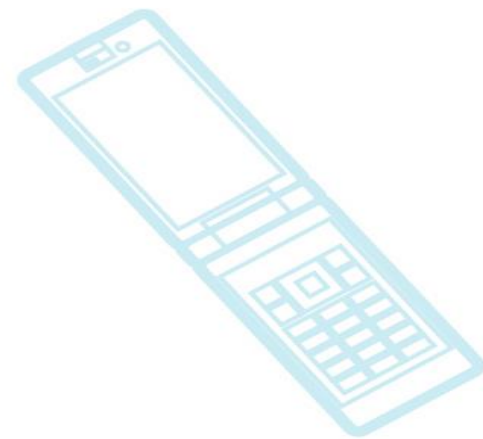
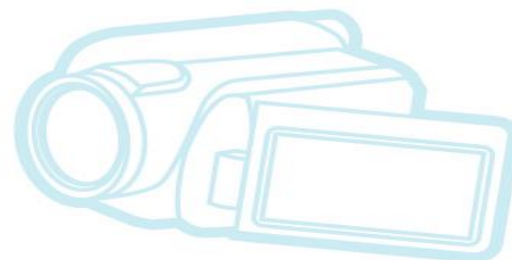
Technology Areas

Industry News

Community

Scorecards

Conclusions





CE Workgroup

# Linux Kernel

- Versions in the last year
  - Pick a few items from each release that are relevant to embedded
    - Lots of things are NOT relevant to embedded (IMHO)
  - Not very good coverage of SoC or driver contributions
- Development Stats





CE Workgroup

# Kernel Versions

- Linux v5.19 – 31 Jul 2022 – 70 days
- Linux v6.0 – 2 Oct 2022 – 63 days
- Linux v6.1 – 11 Dec 2022 – 70 days
- Linux v6.2 – 19 Feb 2023 – 70 days
- Linux v6.3 – 23 Apr 2023 – 63 days
- Linux v6.4-rc3 – Currently on release candidate 3
  - Expect 6.4 release by end of June



# Linux v5.19 (July 2022)

- Support for a.out executable format removed
  - Was deprecated in 5.1
- EROFS read-only filesystem now uses the fscache layer
  - Gives better performance on some systems
- More packet-drop annotations
  - See <https://lwn.net/Articles/885729/>
- Can now embed a 'bootconfig' file directly into the kernel image
- Initial support for the LoongArch CPU architecture
  - See [Documentation/loongarch/introduction.rst](#)



# bootconfig

- Allows passing arguments to kernel as structured key/value pairs
- bootconfig uses a simple text file syntax
- Can be attached to an initrd image, or embedded in the kernel image
- This is useful for passing a large number of tracing options, which would normally make the boot command line too long and complex:
  - e.g. ftrace options, events, filters, actions, probes, fields, etc.
  - Can set up kernel boot-time tracing more easily
- See <https://docs.kernel.org/admin-guide/bootconfig.html>





# Linux v6.0 (October 2022)

- io\_uring supports zero-copy network transmission
  - See <https://lwn.net/Articles/879724/>
- Running Kunit tests now taint the kernel
- Runtime verification system was merged
  - See <https://lwn.net/Articles/857862/>
- CONFIG\_ANDROID config option removed
  - See <https://lwn.net/Articles/899743/>
- printk pull request was rejected this time
  - Developers have been trying to refactor printk for a while now (for RT and other reasons)
  - Back to the drawing board...





CE Workgroup

# Linux v6.1 (December 2022)

- Support for destructive BPF programs
  - Allows a BPF program to crash the kernel (and create a crash dump)
  - See <https://lwn.net/Articles/901284/>
- Experimental support for Rust added
- Improved top-level page for kernel's Documentation
- Support for multi-generational LRU
  - Collects pages into groups by age
  - See <https://lwn.net/Articles/894859/>



CE Workgroup

# Linux v6.2 (February 2023)

- SLOB allocator was deprecated, and the plan is to remove it soon
  - More on this later
- More Rust infrastructure code was added
- Improvements to Squashfs
- New 'rv' tool added to control operation of the runtime verification subsystem



CE Workgroup

# Linux v6.3 (April 2023)

- Lots of old unused Arm board files were removed
- Kernel can be configured with a built-in Dhrystone test
- Default “make V=0” option was been removed
- Minor change to Developer’s Certificate of Origin clarifies that nicknames can be used for contributions
  - As long as your identity is known
  - See <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=d4563201f33a>



CE Workgroup

# Linux v6.4 (June? 2023)

- The SLOB memory allocator was removed
- Some nice documentation added for building the kernel
  - <https://docs.kernel.org/admin-guide/quickly-build-trimmed-linux.html>
- MODULE\_LICENSE() declarations were removed, for code that cannot be built as a module
  - See <https://lwn.net/Articles/927569/>
- User trace events (API fixes) was merged
  - See <https://lwn.net/Articles/927595/>





# LoongArch

- LoongArch is a processor architecture by Chinese company Loongson
- Architecture started in 2002 as a variant of MIPS
  - LoongArch ISA first released in 2021 (so it's new)
- Has incorporated ideas from both MIPS and RISC-V
  - Is not binary compatible with either
- Supports 32-bit reduced instruction set, 32-bit standard, and 64-bit instruction sets
- Linux seems to be the primary OS target for LoongArch
- See <https://en.wikipedia.org/wiki/Loongson>



# Linux 6.1 developer stats

- 13,942 change sets, by 2043 developers (303 new devs.)
  - These are very consistent numbers
- Most active 6.1 developers, by changesets:

Person	Org.	Changesets	Percent	Subsystem Area
Krzysztof Kozlowski	Linaro	221	1.6%	device tree updates
Yang Yingliang	Huawei	169	1.2%	cleanup work in device drivers
Andy Shevchenko	Intel	145	1.0%	various drivers
Johan Hovold	Linaro	145	1.0%	driver fixes and device tree changes
Zhengchao Shao	Huawei	142	1.0%	networking subsystem



# Linux 6.1 developer stats

- Most active 6.1 developers
- By lines of code changed:

Person	Org,	Lines changed	Percent	Subsystem area
Liam Howlett	Oracle	86771	11.3%	maple tree data structure
Frank Min	AMD	56729	7.4%	AMD GPU register definitions
Ping-Ke Shih	Realtek	21352	2.8%	Realtek rtw89 wireless network
Ian Rogers	Google	12558	1.6%	Intel event definitions for perf
Stephen Hemminger	(Retired) was Microsoft	12154	1.6%	Removed support for DECnet

- Note that Stephen made the list for removing lines

Table data:  
<https://lwn.net/Articles/915435/>



# Linux 6.2 developer stats

- 15,536 change sets, by 2088 developers (294 new devs.)
- Most active 6.2 developers, by changesets:

Person	Org.	Changesets	Percent	Subsystem Area
Uwe Kleine-König	Pengutronix	571	3.7%	Fix issue with driver .remove return code
Krzysztof Kozlowski	Linaro	342	2.2%	device tree updates
Johan Hovold	Linaro	178	1.1%	Qualcomm PHY drivers
Andy Shevchenko	Intel	145	1.0%	cleanups in the driver tree
Thomas Gleixner	Linutronix	148	1.0%	core kernel and X86 arch





# Linux 6.2 developer stats

- Most active 6.2 developers
- By lines of code changed:

Person	Org,	Lines changed	Percent	Subsystem area
Ian Rogers	Google	149899	17.0%	Intel PMU event definition files for perf
Ping-Ke Shih	Realtek	31997	3.6%	Realtek rtw89 wireless driver work
Ben Skeggs	RedHat	23004	2.6%	Nouveau graphics driver
Steen Hegelund	Microchip	17222	2.0%	sparx5 network driver
Arnd Bergmann	Linaro	13042	1.5%	Removal of unmaintained drivers

Table data:  
<https://lwn.net/Articles/929582/>



CE Workgroup

# Linux 6.3 developer stats

- 14,424 change sets, by 1971 developers (250 new devs.)
- Most active 6.3 developers, by changesets:

Person	Org.	Changesets	Percent	Subsystem Area
Krzysztof Kozlowski	Linaro	387	2.7%	device tree updates
Dmitry Baryshkov	Linaro	317	2.2%	Qualcomm device drivers
Arnd Bergmann	Linaro	185	1.3%	Removal of old arch and device driver code
Andy Shevchenko	Intel	175	1.2%	Cleanup in the driver tree
Christoph Hellwig	?	167	1.2%	Refactor block and filesystem code



# Linux 6.3 developer stats

- Most active 6.3 developers
- By lines of code changed:

Person	Org,	Lines changed	Percent	Subsystem area
Arnd Bergmann	Linaro	160437	16.4%	Removal of old code
Kalle Valo	Qualcomm	53435	5.5%	Qualcomm WiFi driver
Greg Kroah-Hartman	Linux Foundation	52609	5.4%	Lots of device driver work, removed r8188eu driver from staging
Hans Verkuil	Cisco	28249	2.9%	Removed old media drivers
Cai Huoqing	?	19975	2.0%	Removed lots of DRM drivers

- 4 out of 5 developers made the “top changes” list by removing code!

Table data:  
<https://lwn.net/Articles/915435/>



# Kernel commit log entries

- Number of commit log entries (including merges), per kernel version

Company	git log count	developer count
5.19	16399	2170
6.0	16584	2111
6.1	15109	2117
6.2	16840	2167
6.3	15637	2052
6.4-rc3*	14488*	1836*

- \*we're not done with the 6.4 release cycle yet
- Extracted using 'git log v6.yy..v6.zz --oneline | wc -l' and 'author-stats v6.yy..v6.zz | wc -l'





# Most-Active Organizations, by Area

- Here are the most-active organizations, by kernel area:

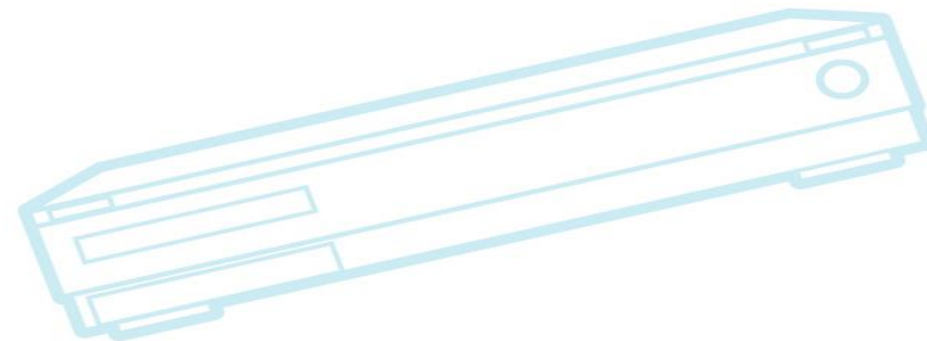
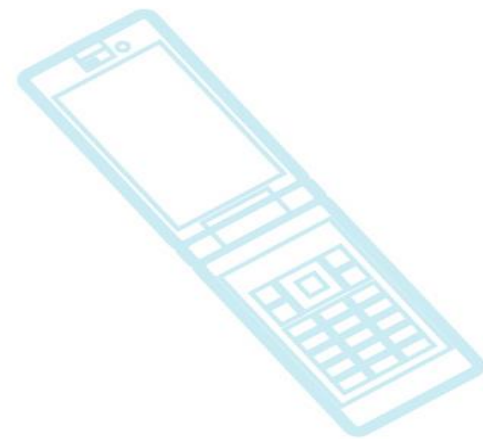
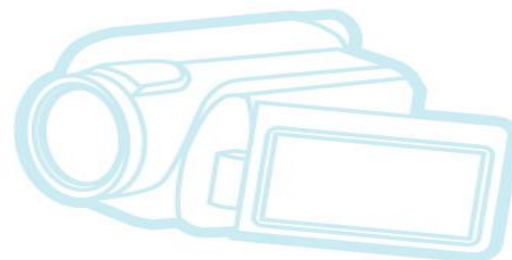
Kernel Area	Organizations with over 5% total contributions from 5.18 to 6.3 (approx. 1 year)
Core kernel	Google, Oracle, Huawei, Red Hat, Meta, Intel
Arch	Linaro, Google, IBM, (Unknown), Intel
Drivers	Intel, AMD, (Unknown), Linaro, (None)
Filesystem & block layer	Red Hat, SUSE, Oracle, Meta, Huawei, (Consultant)
Networking	Intel, Red Hat, Google, Meta, (Unknown), NVIDIA
Documentation	(Unknown), (None), Google, Intel, Red Hat, Meta, Loongson



CE Workgroup

# Outline

Linux Kernel  
**Technology Areas**  
Industry News  
Community  
Scorecards  
Conclusions





# Technology Areas

- Architectures
- Bootloaders
- Core Kernel
- Filesystems
- Languages
- Networking
- Security
- Testing
- Tools
- Toolchains
- Tracing
- System Size
- Build Tools and Distros



# Architectures

- LoongArch support recently added (v5.19 and several 6.x kernels)
- Support for many old ARM boards was recently removed
- Proposed removal of old Architectures
  - Super-H
  - Itanium
- Community discussion: How long to keep code in kernel?
  - Answer seems to be: as long as it doesn't impose a burden on maintainers for other areas
  - Or, someone cares about latest kernel enough to step up and maintain it
- See <https://lwn.net/Articles/920259/>





CE Workgroup

# Bootloaders

- U-Boot
  - Now supports loading images over HTTP
    - Previously only supported the UDP protocol
    - Could only use NFS or TFTP as servers
    - Now can download kernel and other images (dtb, initrd, etc.) from a web server
    - See <https://www.linaro.org/blog/http-now-supported-in-u-boot/>
- Snagboot
  - See next page



# snagboot

- Snagboot is a set of tools that can help boot and install images on boards that fail to boot
- Produced by Bootlin
- Consists of
  - snagrecover – to initialize memory and run your bootloader
  - snagflash – to flash a working system image using either DFU, USB Mass Storage, or fastboot
- Works on several different boards, and replaces proprietary tools
- See <https://bootlin.com/blog/releasing-snapboot-a-cross-vendor-recovery-tool-for-embedded-platforms/>



# Core Kernel

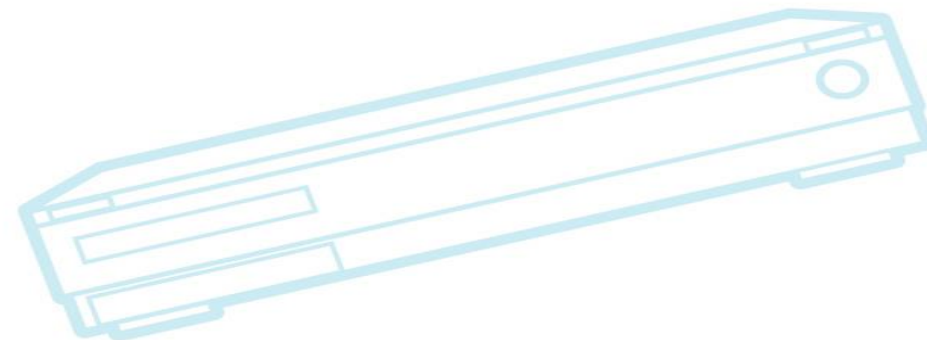
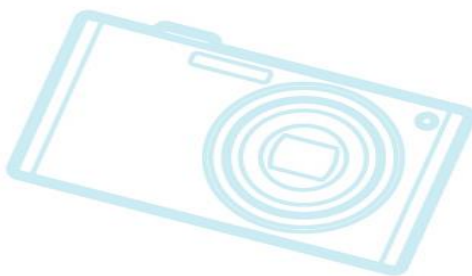
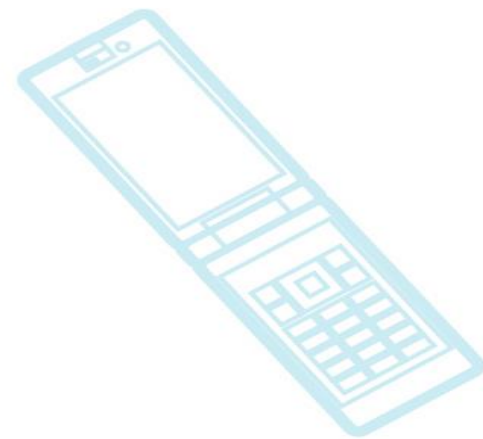
- Work being done on io\_uring\_spawn
  - Replaces fork/exec
    - Is more efficient
    - fork/exec almost always discards the parent's code immediately
  - Can do IORING\_OP\_CLONE followed by IORING\_OP\_EXEC in the kernel
  - 6-10% faster than vfork() and 30+% faster than posix\_spawn()
  - See <https://lwn.net/Articles/908268/>



CE Workgroup

# Filesystems and I/O

- Lots of tweaks to existing filesystems
  - Not much specific to embedded



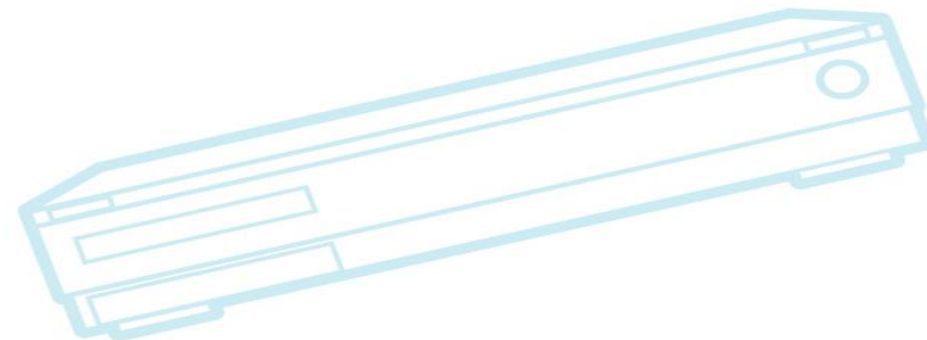
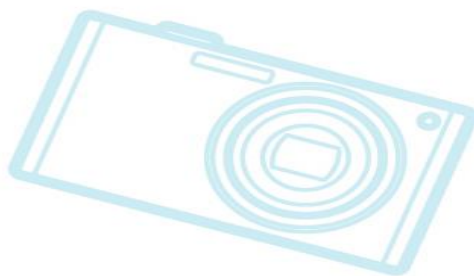
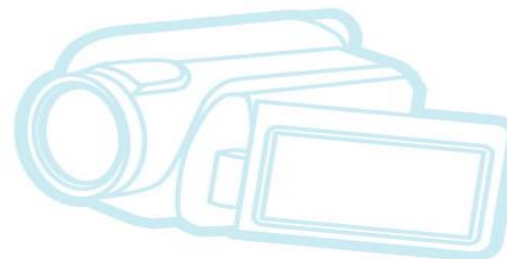
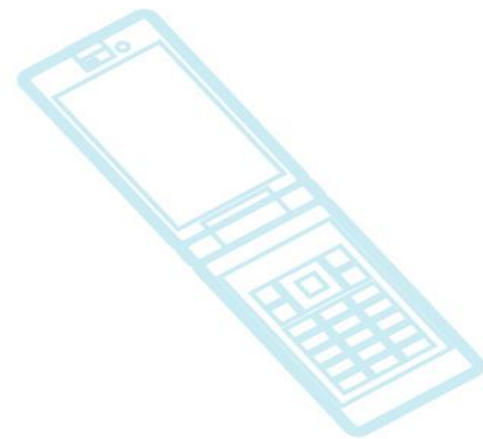




CE Workgroup

# Languages

- Micro-Python
- Python
- Rust





CE Workgroup

# Micro-Python

- Version of python specifically targeted at microcontrollers and embedded use cases
- Version 1.2 released in April, 2023
- New Features:
  - New package manager (mip) can install packages from the micropython-lib repository
  - Reduced code size
    - Developers can pre-compile modules
  - Add support for many new boards
    - Support for Zephyr
    - Can target WASM (web assembly), to run in Python apps in browsers
- See <https://lwn.net/Articles/931051/>



CE Workgroup

# Python

- Python 3.11.3 released in April, 2023
- Python 3.11 has a lot of improvements
  - Better tracebacks
    - Fine-grained error locations in tracebacks
  - Faster (!!)
    - Claims a 1.22x speedup on the standard benchmark suite
    - Up to 10-60% faster than Python 3.10
  - Exception groups and except\*
  - Typing improvements
  - See <https://discuss.python.org/t/python-3-11-0-final-is-now-available/20291>



CE Workgroup

# Python programs debugged using AI

- “Wolverine” project debugs Python programs using AI
- Wolverine runs the program, and performs “automatic healing”
- When a crash is detected, it uses GPT-4 to:
  - Analyze the failure
  - Generate and apply a bugfix
  - Re-run the program, to continue to detect problems
- See <https://arstechnica.com/information-technology/2023/04/developer-creates-self-healing-programs-that-fix-themselves-thanks-to-gpt-4/>





# Rust

- Rust support continues to go into the mainline kernel
  - But it has not been used for a “real” driver yet
- My own impression:
  - Rust is being used more and more throughout the industry
    - ex: KataOS = secure operating system written by Google
    - See <https://www.phoronix.com/news/Google-KataOS>
- Rust 1.71 will likely support the musl C library
- Google announces their results analyzing Rust crates
  - See <https://opensource.googleblog.com/2023/05/open-sourcing-our-rust-crate-audits.html>



CE Workgroup

# Networking

- New internal function to provide reason for a packet drop (5.17)
  - Helps administrators determine reason for networking issues
    - But only 63 of 4000 kfree\_skb calls converted so far (as of 5.17)
  - See <https://lwn.net/Articles/885729/>
  - Lots more instrumentation has been done over the last year
    - About 300 sites instrumented now



CE Workgroup

# Real-Time

- PREEMPT\_RT status
  - Sleeping locks was mainlined (v5.16)
  - Patches have been going in continuously – through 6.4
    - Although some stalled (like the printk refactor)



CE Workgroup

# PREEMPT\_RT - What's left

- What's left in PREEMPT\_RT patches out of mainline:
  - last year (2022):
    - About 1300 lines of code, affecting 92 files (in 51 patches) (!!)
  - this year (2023):
    - About 4500 lines of code, affecting 100 files (in 84 patches)
      - Number of lines could be off due to printk work-in-progress
    - Some changes to the printk, 8250 serial driver, and the core scheduler (and other places)
  - People are **still** anxious for Linux RT without having to apply a patch
- See <https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/5.19/patches-5.19-rc1-rt1.tar.gz>

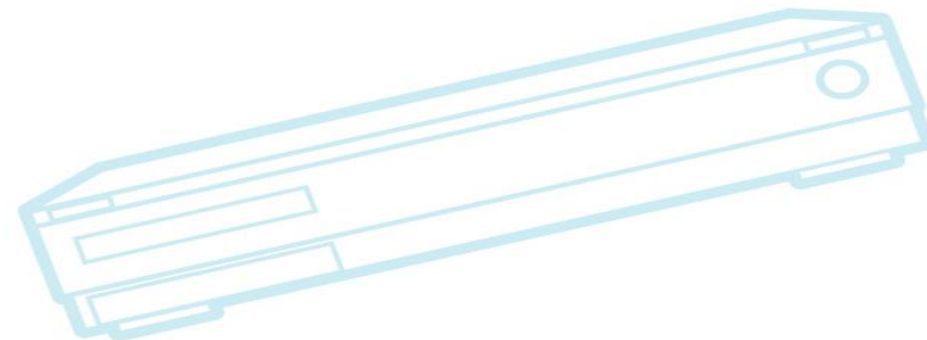
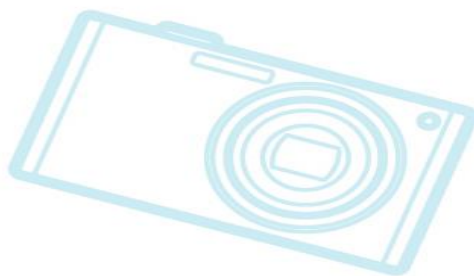
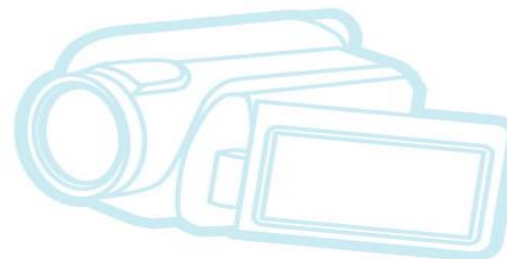
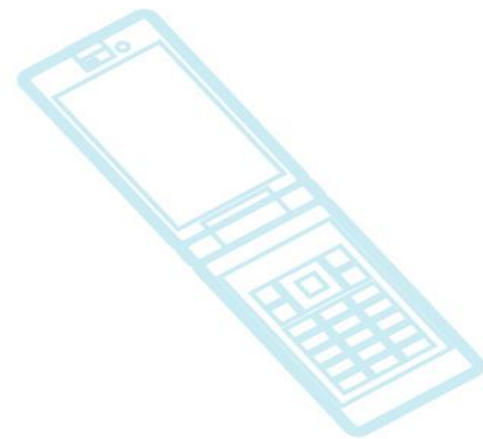




CE Workgroup

# Security

- Kernel hardening continues





CE Workgroup

# System Size

- SLOB memory allocator removed from kernel (v6.4)
  - Not enough people using it (or reporting that they used it)
  - Was deprecated in v6.2
    - IMHO, that wasn't enough time for people to object
  - Don't move to SLAB, as that's planned for deprecation also
  - Should move to SLUB, and use CONFIG\_SLUB\_TINY
- Bloaty McBloatface size profiler tool
  - Tool by Google to analyze size of binary images
  - Does deep inspection of ELF binaries
  - Can show comparison between two binaries
  - See <https://github.com/google/bloaty>
  - Was open-sourced in 2016 and I just found it this year!



# nolibc

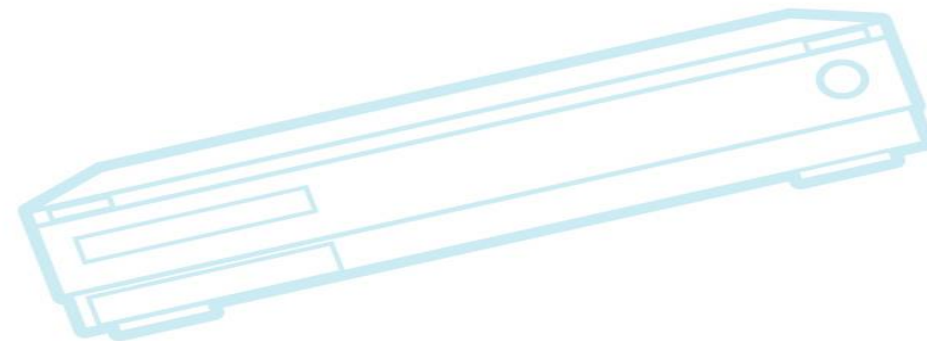
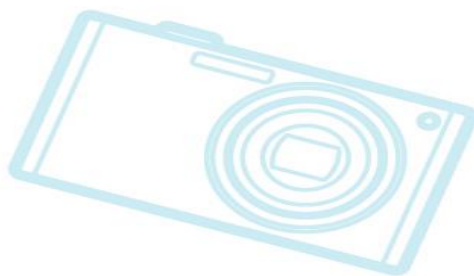
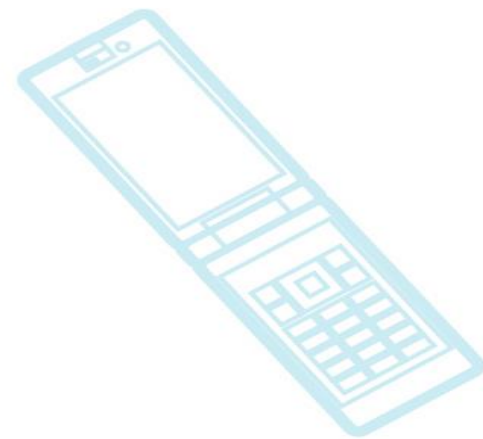
- Is a minimal C-library replacement shipped with the kernel
  - Originally created for kernel testing
  - Consists of a set of .h files, which create macros to replace common C library functions
- Allows creation of a very small static binary
  - For some binaries, the statically linked binary using nolibc is smaller than the binary using libc (using dynamic libraries)
    - e.g. hello-libc size=1159 bytes, hello-nolibc= 1100 bytes
- Lots of issues to look out for, but an interesting approach to create minimal binaries in Linux
- See <https://lwn.net/Articles/920158/>



CE Workgroup

# Testing

- Systems
- Suites of tests







CE Workgroup

# Test Systems

- 0-Day – reports bugs at time of patch submission
- KernelCI – has added kselftest git repo to list of trees it tests
- Syzbot – always producing more fuzzing failure cases
- CKI – providing many reports to upstream
- LKFT – providing many reports to upstream



CE Workgroup

# Test Suites

- LTP
  - Latest release: 20230516 (May 27, 2022)
    - New test and fixes to tests
    - Added new “runltp-ng” – test execution manager
      - Can execute tests on a separate machine
    - See <https://github.com/linux-test-project/ltp/releases>
- kselftest
  - Lots of tests and test improvements for bpf, mm, resctrl, hid, xsk, networking, x86 features, powerpc, KVM, landlock and more
- Kunit
  - kunit tool improvements



CE Workgroup

# Toolchains - GCC

- GCC 13.1 released April 26, 2023
  - See <https://gcc.gnu.org/gcc-13/changes.html>
  - LTO (link-time optimizations) has been improved
  - Can now emit diagnostics in the SARIF and gcc's own JSON-based format
    - SARIF = Static Analysis Results Interchange Format
  - Support for C23 features
  - LoongArch support improvements
  - Improvements in the static analyzer



CE Workgroup

# Toolchains - LLVM

- LLVM 16.0.0 released March 17, 2023
  - See <https://releases.lvm.org/16.0.0/docs/ReleaseNotes.html>
- People are using it for whole distributions, not just kernel
- See presentation: “Experiences of OS distributions using LLVM as their main toolchain”
  - By Bernhard Rosenkranzer at 2022 European LLVM Dev. meeting
  - <https://youtu.be/h9xg8Y8bylg>





# Tools

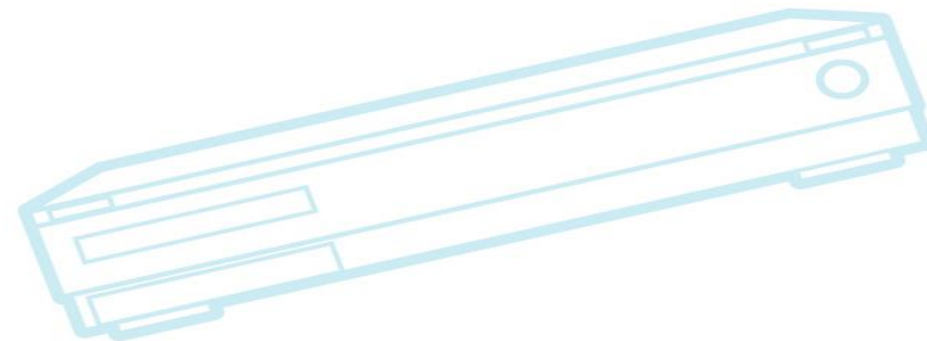
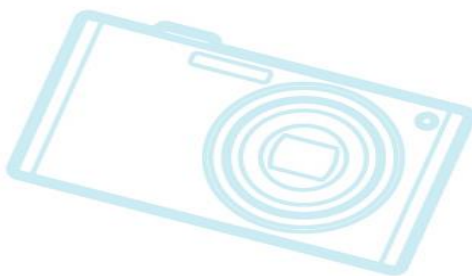
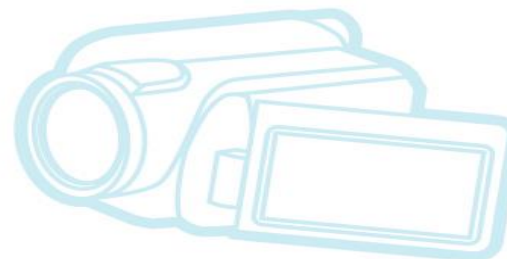
- **b4**
  - Is a tool for retrieving patches from mailing lists
  - But can also help contributors:
    - Create and manage patch series and cover letters
    - Track and auto-reroll series revisions
    - Display range-diffs between revisions
    - Apply trailers received from reviewers and maintainers
    - Submit patches without needing a valid SMTP gateway
- **b4 submission of kernel patches**
  - It is now possible to contribute a patch without using email!
  - However, it's a good idea to be subscribed to relevant lists in order to respond to comments
  - See <https://people.kernel.org/monsieuricon/sending-a-kernel-patch-with-b4-part-1>



CE Workgroup

# Tracing

- Something is always going on with perf
  - New features seem to be added every release
- eBPF is now used for tracing





# Build Tools and Distros

- Yocto Project
  - BWM joined Yocto Project
    - See [https://www.theregister.com/2022/07/20/yocto\\_project\\_4/?td=keepreading](https://www.theregister.com/2022/07/20/yocto_project_4/?td=keepreading)
  - Latest version = 4.2 (Mickeldore), released May, 2023
    - Kernel 6.1 and 350+ recipe upgrades
    - Improved memory and disk usage
    - Improved recipe parsing time
    - Now requires Python 3.8
    - Bunch of other changes
    - See release notes:
      - <https://docs.yoctoproject.org/next/migration-guides/release-notes-4.2.html>



CE Workgroup

# Kernel community

- Linus now uses scripts to check language in his emails
  - When Linus quotes other people's email, sometimes the script catches a word or phrase that Linus did not write
  - Linus has kind asked that other developers "please don't call each other morons on the mailing lists"
  - See <https://lwn.net/Articles/908466/>
- Slight change to DCO (Developer Certificate of Origin)
  - Can use "known identity" instead of "real name"
  - Still precludes anonymous contributions, but allows nicknames
  - See <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=d4563201f33a>

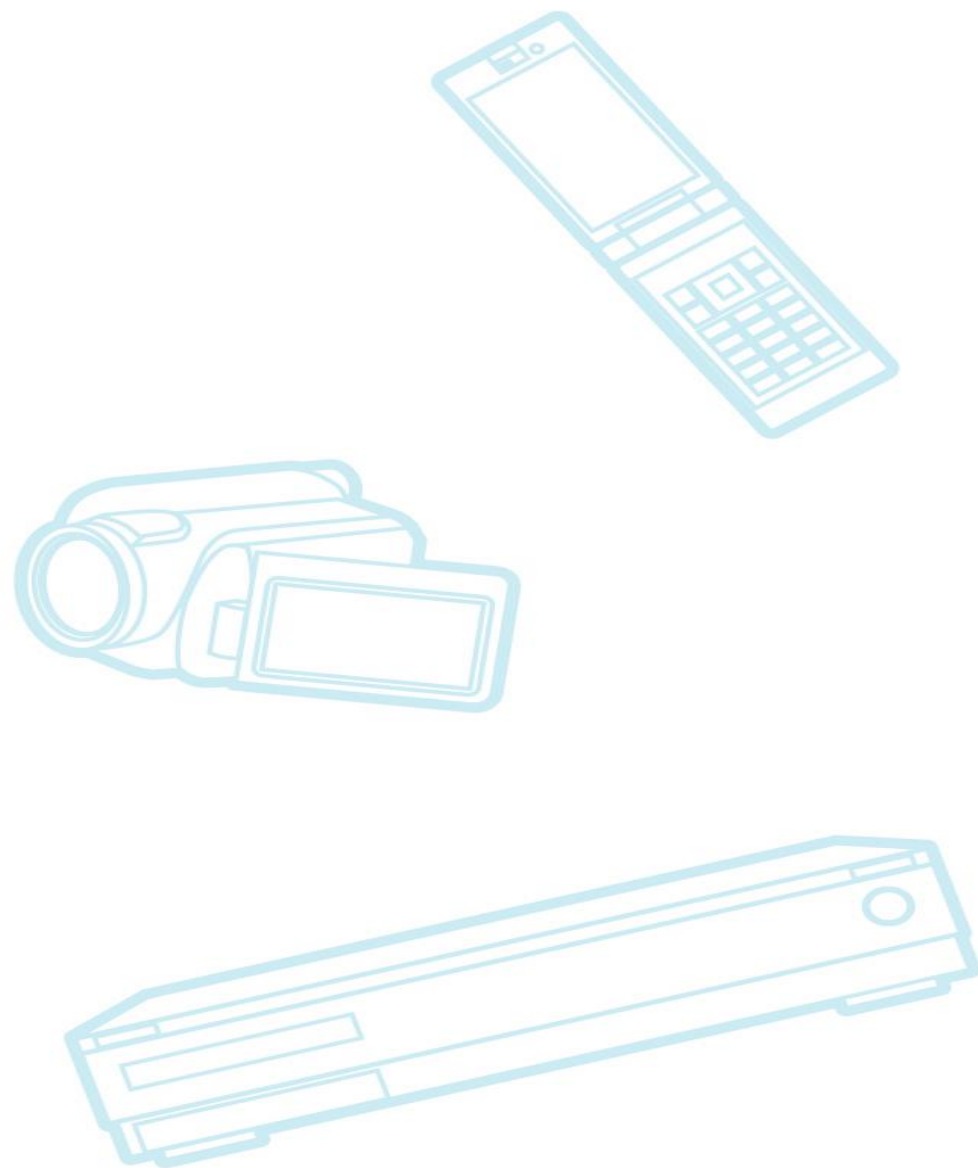




CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
**Industry News**  
Community  
Scorecards  
Conclusions

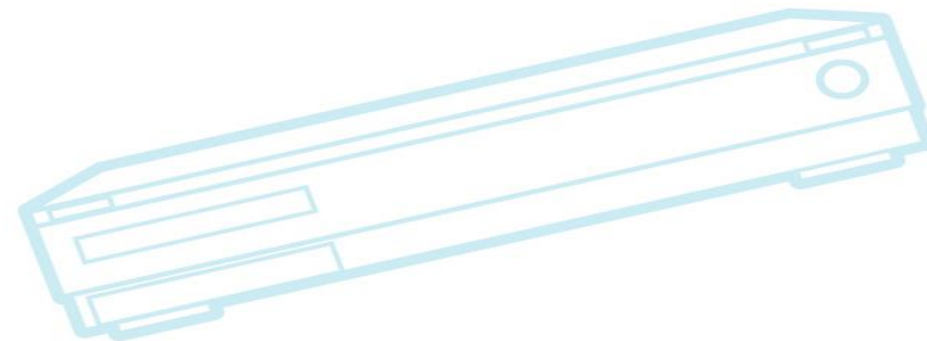
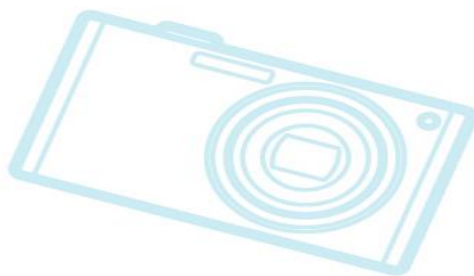
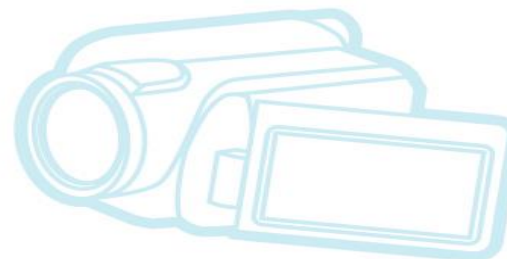




CE Workgroup

# Industry News

- Legal issues
  - SFC sues Microsoft over github co-pilot
- Intel and ARM agreement
- NASA and RISC-V
- Miscellaneous





CE Workgroup

# SFC sues Microsoft over github co-pilot

- SFC alleges that use of GPL code in training data for co-pilot is a GPL violation
  - Argument is that software was used and license terms are being violated
  - SFC gave some examples where code produced by co-pilot exactly matched GPL code
- Similar in spirit to other lawsuits for StableDiffusion
- Raises questions about copyright interaction with training data for ML models
- See <https://www.theverge.com/2022/11/8/23446821/microsoft-openai-github-copilot-class-action-lawsuit-ai-copyright-violation-training-data>



# Intel Foundry and ARM agreement

- Intel and ARM announce collaboration on designs using Intel's 18A process
  - 18A is farthest item in a long-term roadmap by Intel for continued shrinking of processor features
  - 18 Angstroms = 1.8 nanometers
- Quote: "By unlocking Arm's leading-edge compute portfolio and world-class IP on Intel process technology, Arm partners will be able to take full advantage of Intel's open system foundry model, which goes beyond traditional wafer fabrication to include packaging, software and chiplets."
- See <https://www.intel.com/content/www/us/en/newsroom/news/intel-foundry-arm-announce-multigeneration-collaboration-leading-edge-soc-design.html>





# NASA and RISC-V

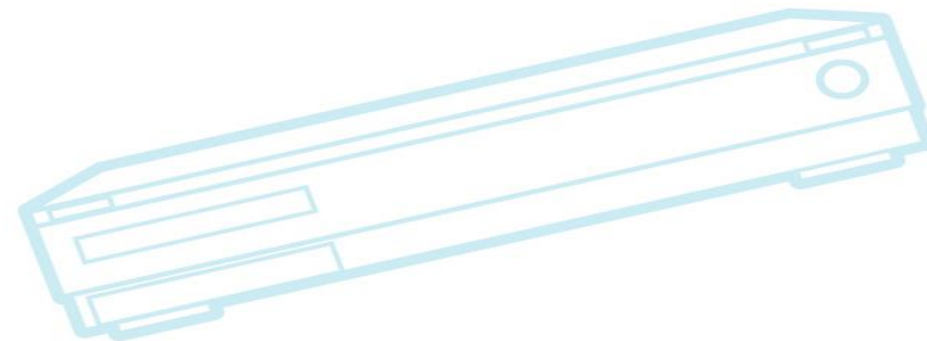
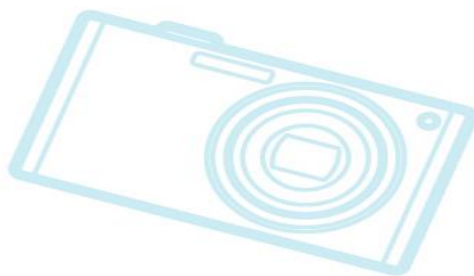
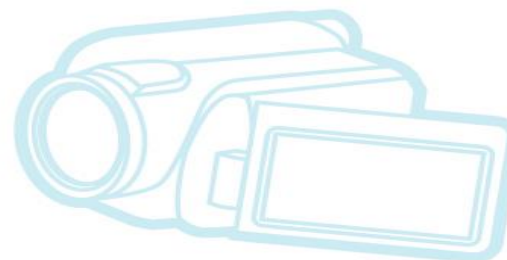
- NASA JPL (Jet Propulsion Lab) have selected Microchip to design and produce a 12-core RISC-V processor
- For the “High Performance Spaceflight Computer”
- 8 RISC-V cores with vector processing extensions (in 2 clusters) + 4 general purpose RISC-V cores
- Is designed for high performance and fault tolerance
- Maybe be used outside of space applications
- See <https://www.eejournal.com/article/nasa-recruits-microchip-sifive-and-risc-v-to-develop-12-core-processor-soc-for-autonomous-space-missions/>



CE Workgroup

# Industry News

- Miscellaneous
  - Lennart Poettering has a new job
  - Interesting cases of embedded Linux





CE Workgroup

# Miscellaneous

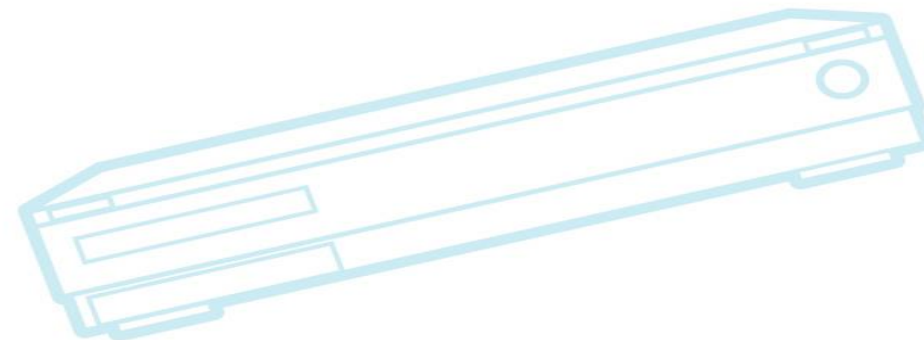
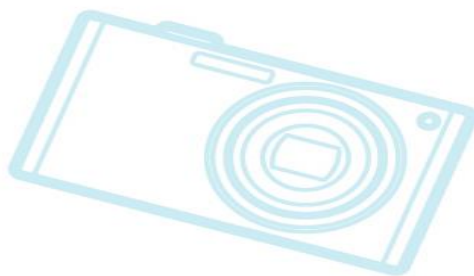
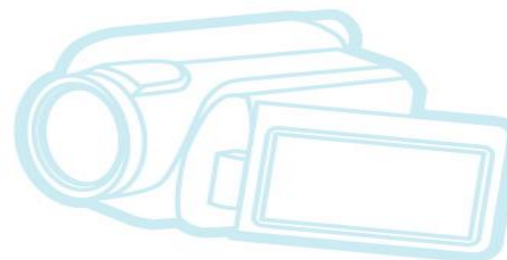
- Lennar Poetering has a new job
  - Lennart is the author of systemd
  - Left Red Hat, and is now working for Microsoft (as of July 2022)
  - Lennart is still working on systemd
  - Microsoft has hired a number of Open Source and Linux leaders in the last few years
    - MS OSS folks may be associated with Azure, or Windows Subsystem for Linux
- See [https://www.phoronix.com/scan.php?page=news\\_item&px=Systemd-Creator-Microsoft](https://www.phoronix.com/scan.php?page=news_item&px=Systemd-Creator-Microsoft)



CE Workgroup

# Interesting embedded Linux uses

- Mars Ingenuity helicopter
- Starlink satellite constellation







CE Workgroup

# Mars Helicopter - Ingenuity





CE Workgroup

# Mars Helicopter

- Mars Ingenuity Helicopter landed in February, 2021 on Mars
- Performed tests and demonstrations in April & May (2021)
  - First 5 flights were part of “Technology Demonstration”
- After demo, NASA created a plan for continued flights
- Is still flying...
  - Has performed 51 flights so far
- Updates:
  - Autonomous landing site selection





CE Workgroup

# Ingenuity Helicopter Update (May 2023)

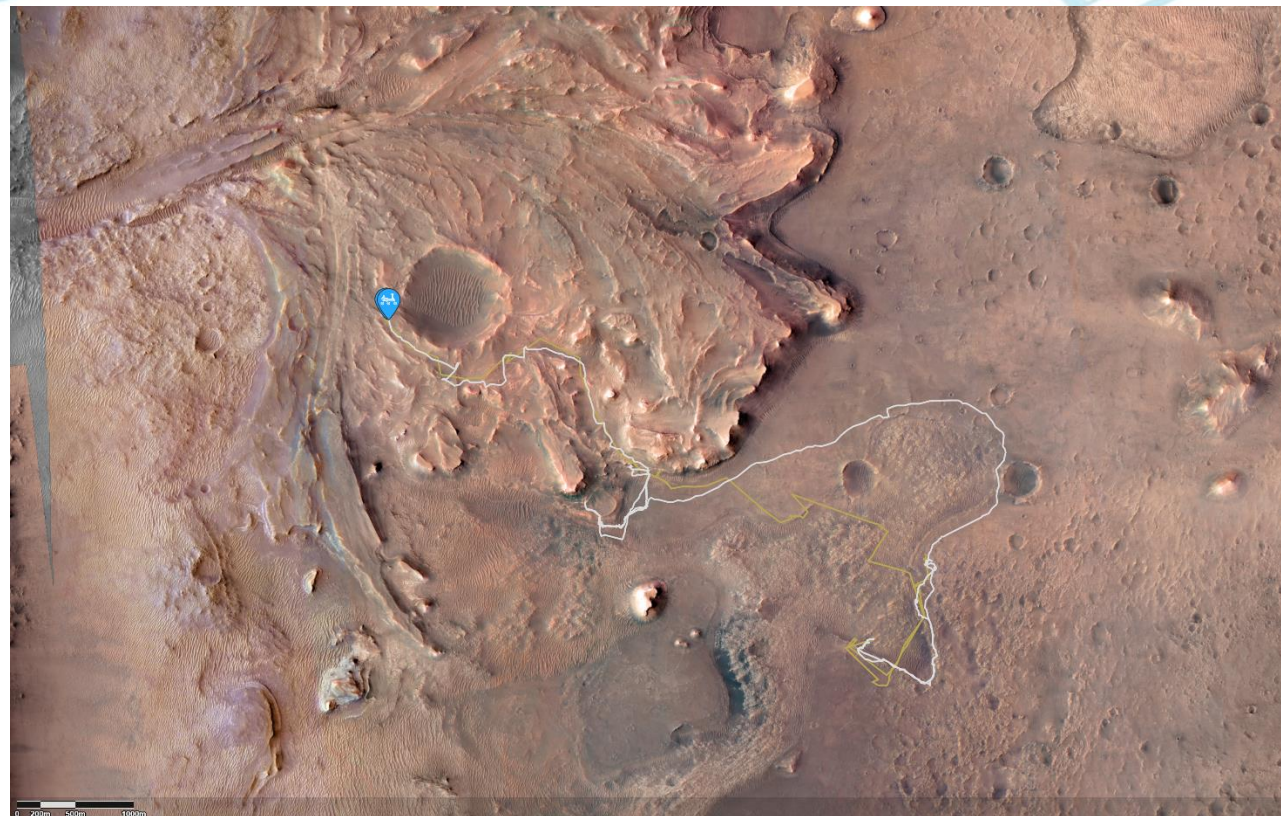
- Autonomous Landing Site Selection
  - NASA uploaded a new Landing hazard mitigation system
  - Used first in flight 39
  - Detects the slope of the landing area, and any debris that might interfere, and adjusts landing position
- Scouting for Perseverance
  - Provides pictures of areas of interest, and potential navigation concerns for the rover Team
  - Now off the crater floor, terrain is more rugged, and helicopter may land somewhere outside of communications range
  - Flights 41-46 consisted of keeping Ingenuity ahead of the rover
    - Canyon was too narrow for Ingenuity to safely pass the rover if it fell behind





CE Workgroup

# Ingenuity flights on Mars







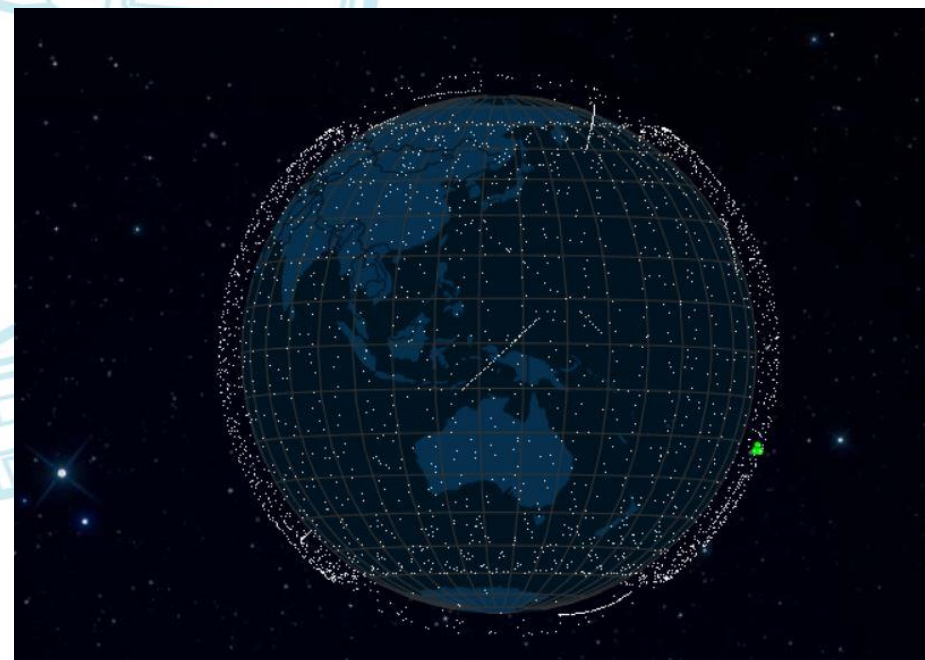
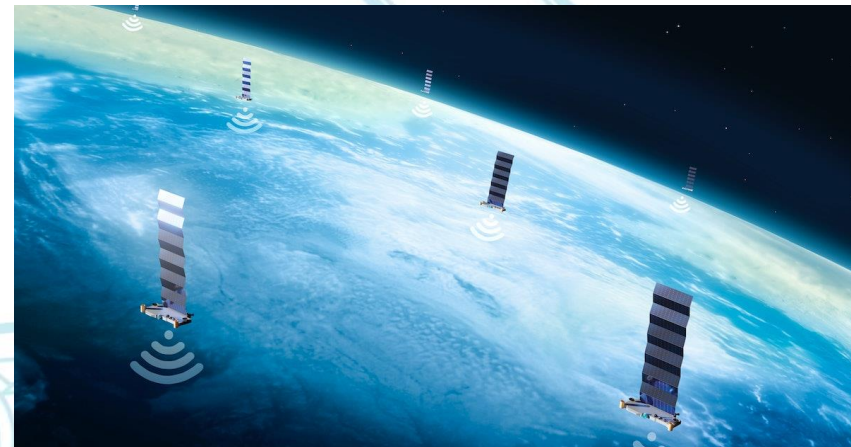
# Sources for Mars helicopter

- Talk by Tim Canham at ELC 2021
  - Slides: [https://elinux.org/images/5/5a/1.\\_TIMOTHY\\_CANHAM.pdf](https://elinux.org/images/5/5a/1._TIMOTHY_CANHAM.pdf)
  - Video: [https://youtu.be/0\\_GfMcBmbCg](https://youtu.be/0_GfMcBmbCg)
- <https://mars.nasa.gov/technology/helicopter/>
- [https://en.wikipedia.org/wiki/Ingenuity\\_\(helicopter\)](https://en.wikipedia.org/wiki/Ingenuity_(helicopter))
- <https://thenewstack.io/how-the-first-helicopter-on-mars-uses-off-the-shelf-hardware-and-linux/>
- <https://www.pcmag.com/news/4-android-smartphones-with-as-much-power-as-nasas-mars-helicopter>



# Starlink Satellite constellation

- SpaceX uses Linux in their rockets, space capsules and satellites
- Each Starlink satellite uses over 60 processors, each one running Linux
  - Uses clusters for fault tolerance
    - Voting algorithms
    - Sub-component reboot capabilities
    - Redundant failover
- There are now (as of May 2023) over 4000 Starlinks satellites currently in orbit
  - <https://satellitemap.space/> is quite interesting!!

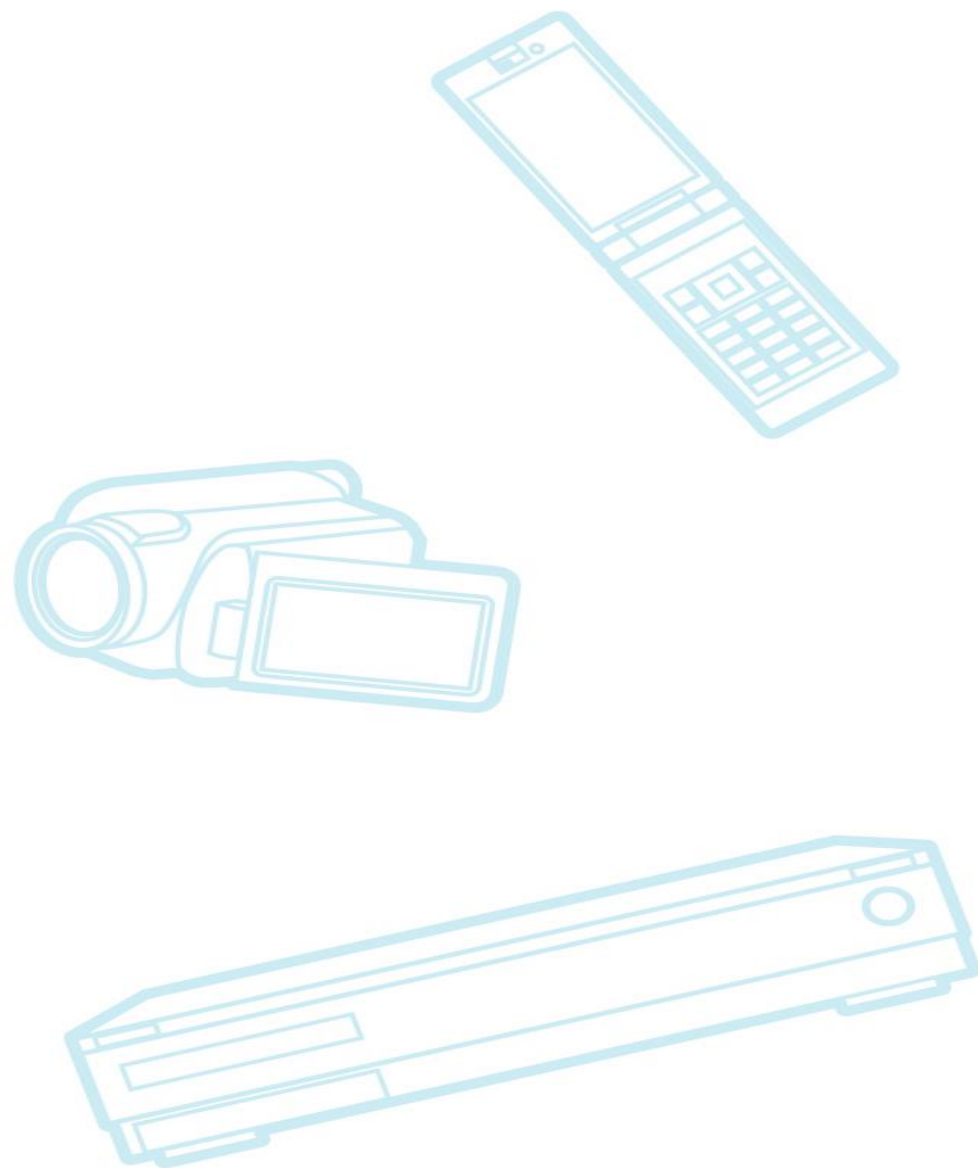




CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Industry News  
**Community**  
Scorecards  
Conclusions



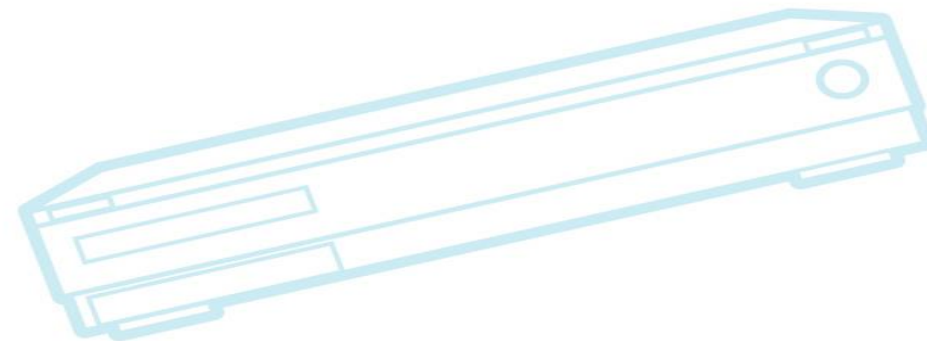
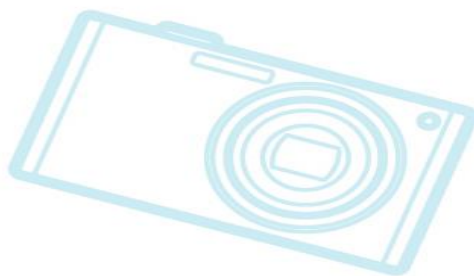
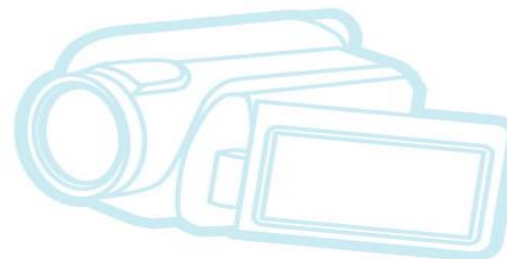
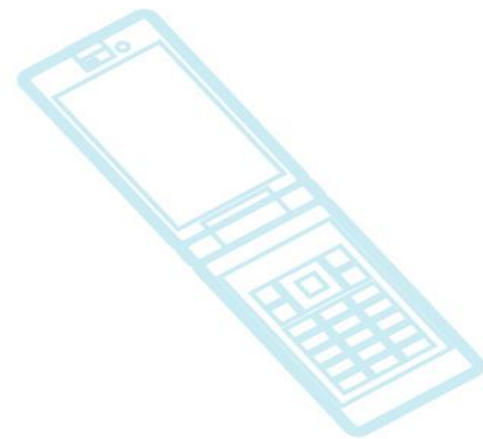




CE Workgroup

# Community

- A sad note
- Conferences
- Elinux wiki
- Trade Associations and Projects







CE Workgroup

# A sad note about Wolfgang Denk

- On October 14, 2022, Wolfgang Denk died
- Wolfgang was the founder of Denx Software Engineering
- A pioneer in the field of embedded Linux
  - The creator “Das U-Boot”
    - With a sense of humor
  - Early work in realtime solutions for Linux
- A champion for Open Source
- Wolfgang will be missed





CE Workgroup

# Conferences

- Embedded Linux Conference
  - Is now only once a year
    - Alternates between Europe and North America
    - So, effectively only once per region every two years
    - May decide to fill in with regional events or Plumbers
  - Sometimes in Embedded Open Source Summit, and sometimes in Open Source Summit (North America)
  - Consistently in the spring (April/May)
  - Next one = June 28-30 in Prague, Czech Republic
  - 2024 = April 15-19 in Seattle, USA



# Conferences (cont.)

- Japan Jamboree
  - Not sure if there will be more after this one (#80)
- Embedded Recipes
  - September 21-22, 2023 in Paris
- Linux Plumbers
  - Could have an embedded microconference
  - November 13-15, 2023 in Richmond, Virginia, USA
- FOSDEM
  - Has an embedded track
  - February 3,4 2024 (not confirmed) in Brussels





CE Workgroup

# Elinux wiki

- Losing funding for our elinux.org administrator (Bill Traynor)
- Site is still used for:
  - Materials for embedded Linux development boards
  - Some academics use it for coursework
  - Event materials: slides and links to videos for ELC
- Other areas of the site are often out-of-date
- I feel like the site is underutilized for sharing information
- What to do next....? (Any volunteers?)





# Trade Association and Projects

- Linaro – does ARM upstreaming and projects
  - Still going strong (? apparently)
- Robot Operating System (ROS) – handles robotics vertical
- Android Open Source Project (AOSP) – handles mobile phone vertical
  - And there are others (e.g. LineageOS, PinePhone, etc.)
- Buildroot, OpenWRT – build systems for embedded Linux and routers



CE Workgroup

# Linux Foundation projects

- Linux Foundation
  - Core Infrastructure Project (CIP) – handles support longevity
  - ELISA – handle issues with safety certification and standards
  - OpenChain – handles issues with supply chain
  - SPDX – handles licensing issues and SBOMs
  - Automotive Grade Linux (AGL) – handles automotive vertical
  - KernelCI – handles automated testing (for upstream)
  - Yocto Project – handles common build system for embedded
  - DroneCode – handles drone vertical
  - Core Embedded Linux Project – is shutting down



# Core Embedded Linux Project

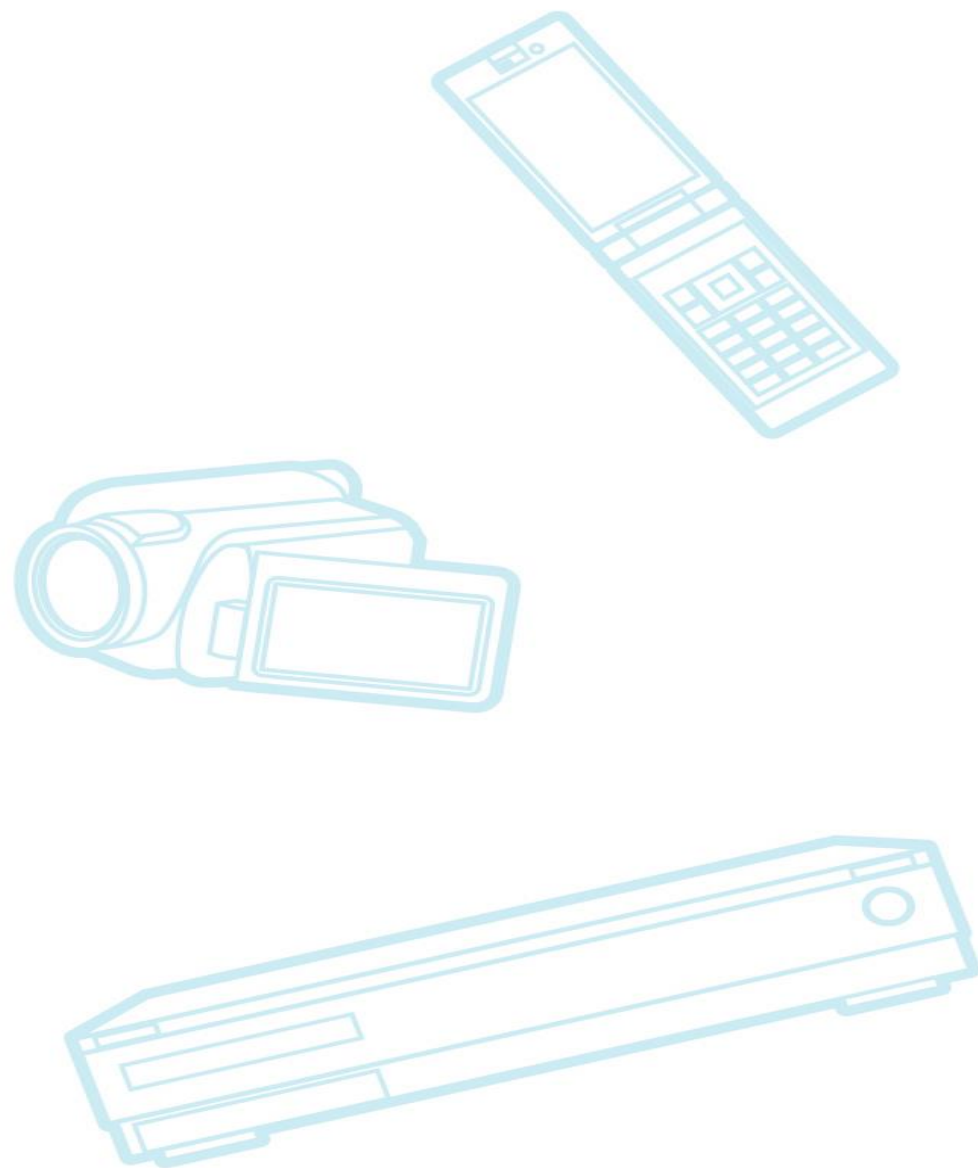
- History:
  - Started as Consumer Electronics Linux Forum in 2003
  - Migrated to Embedded Linux Workgroup in the LF in 2011
  - Became Core Embedded Linux Project in 2015
- Funded upstreaming of various technologies
  - Linux-tiny patches, squashfs, realtime work, power management features, etc.
- Started the Embedded Linux Conference
- Sponsored the eLinux wiki for many years
- Now, plan to end in July 2023 (after 20 years)



CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Industry News  
Community  
**Scorecards**  
Conclusions



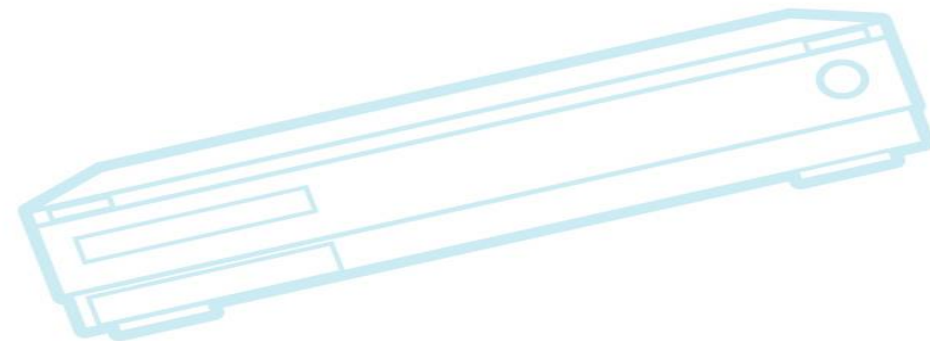
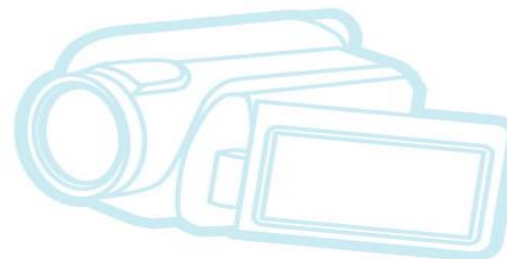
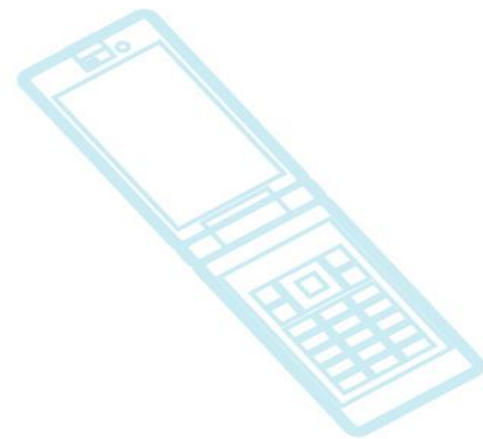




CE Workgroup

# Scorecards

- Are we done yet?
- Well, where are we?
- Scorecards for:
  - Technology
  - Development

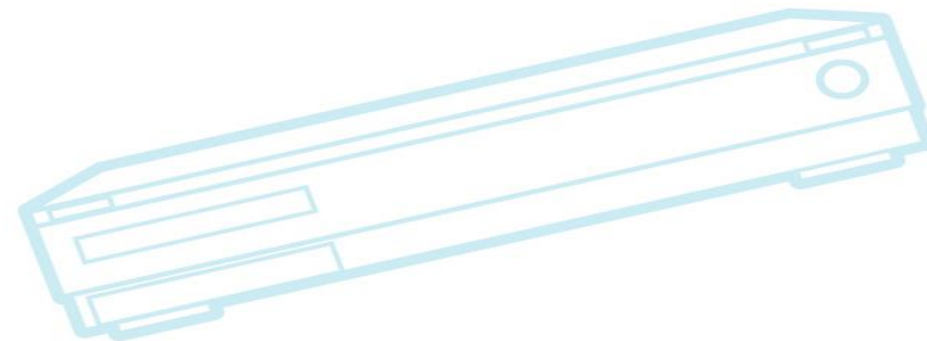
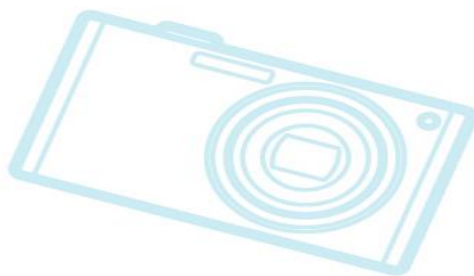
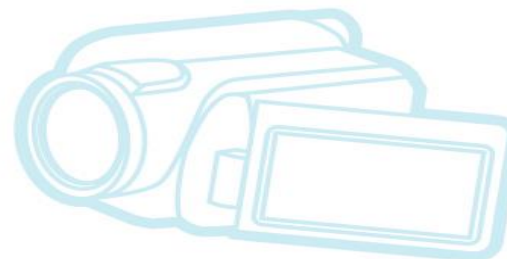




CE Workgroup

# Technology Scorecard

- Original focus areas:
  - System size
  - Boot time
  - Power management
  - Realtime
  - Security





CE Workgroup

# Technology Scorecard

- Based on contributions in the last few years:
- System size – done
- Boot time – done
- Power management – done
- Realtime – done
- Security – in progress
- Congratulations everyone!! We did it!!
  - World(s) domination achieved!



# Technology Scorecard (reality and explanations)

- System size – **done**
  - Lower limit is about 16M, and it's not getting any better
  - Alas, Linux will never be on 1-cent processors
    - The 10 trillion IoT sensors will be running something else (darn)
- Boot time – **done**
  - Cold boot boot time reduction largely abandoned in mainline
    - People do heroic special-casing to get low boot times, when required
  - Most products use either suspend/resume or low-power idle





CE Workgroup

# Technology Scorecard (explanations)

- Power Management – **done??**
  - governors, frequency scaling, power domains, power qos, and power scheduling features are all upstream
  - It requires SoC and board support (e.g. driver pm integration) for it to work
  - It's now mostly a BSP (Board Support Package) problem



CE Workgroup

# Technology Scorecard (explanations)

- Realtime – **done??**
  - **PREEMPT\_RT** code is (*almost all*) upstream!!
  - But it requires ongoing maintenance to avoid changes that damage realtime performance
- Security – **in progress**
  - kernel hardening, handling security reports, Rust drivers
  - Alpha/Omega project



CE Workgroup

# Real Technology Scorecard

- System size – done
- Boot time – done
- Power management – done??
- Realtime – done??
- Security – in progress

Recognise now that all of these are “holistic”.

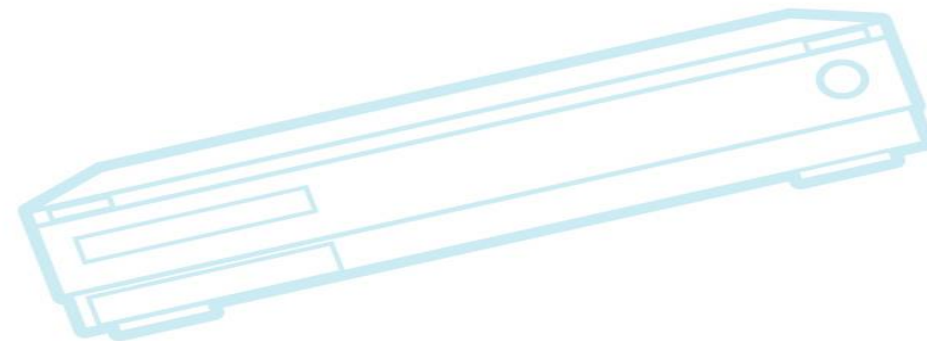
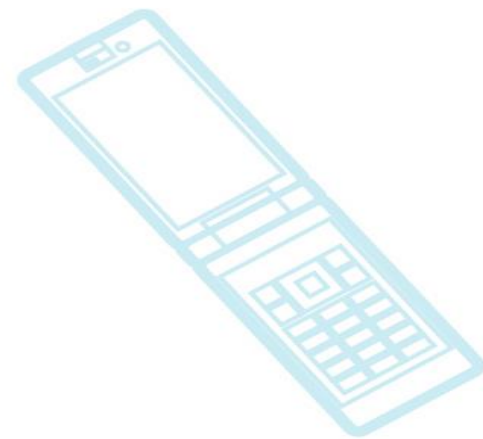
- All of these require pervasive, constant maintenance, distributed throughout the code base



CE Workgroup

# Development Scorecard

- Build systems/Distros - good
- Training/Consulting - good
- Toolchains – good
- Debugging capabilities – good
- Languages – good
- Test Systems – in progress
- Hardware support – in progress



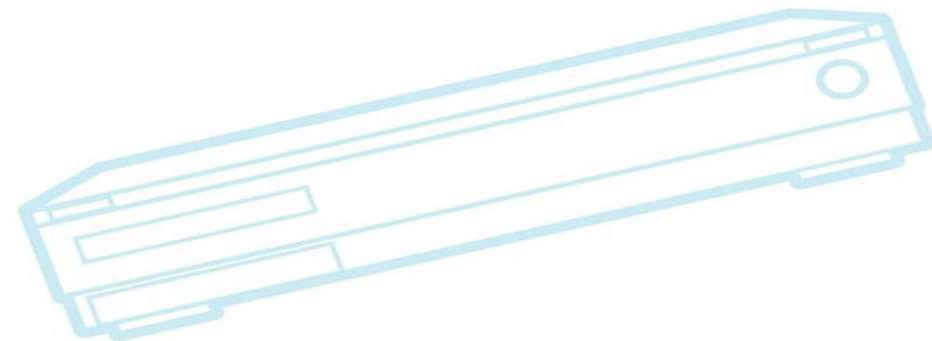
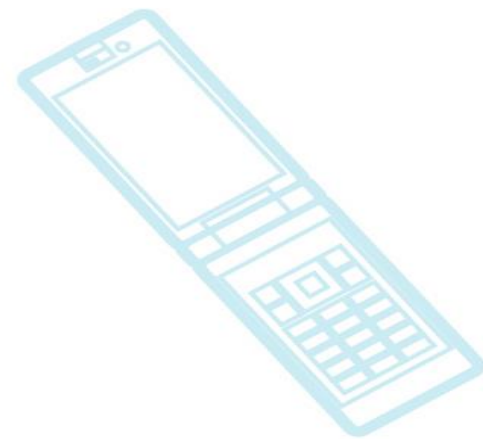
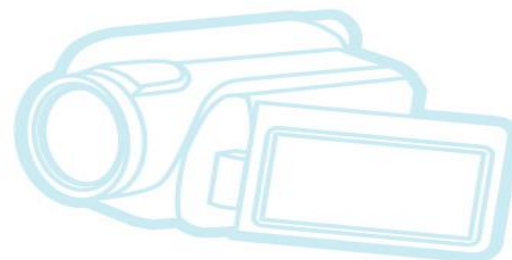




CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Industry News  
Community  
Scorecards  
**Conclusions**





# Conclusions

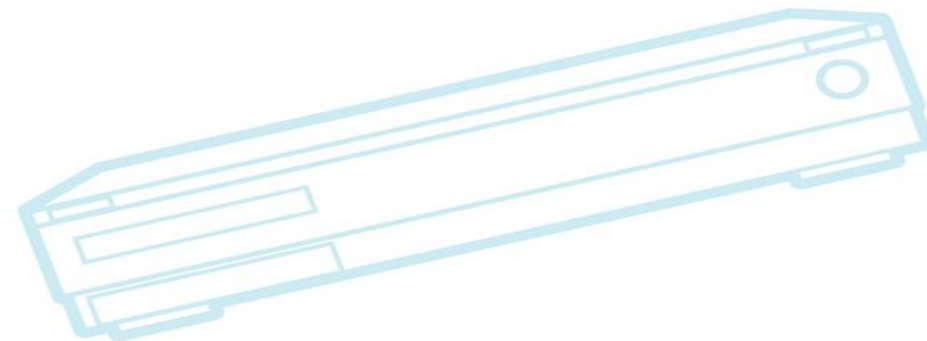
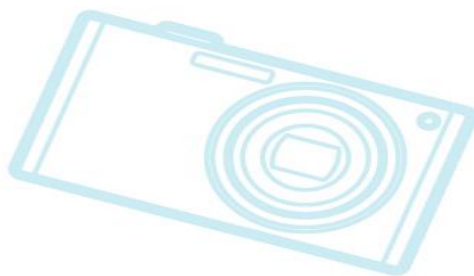
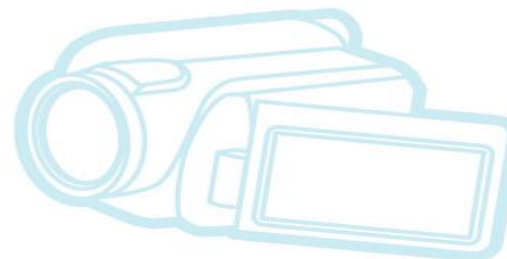
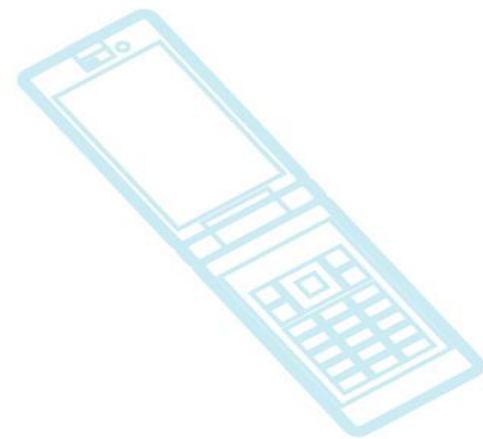
- Overall – we're doing pretty well
  - Embedded Linux is widely deployed and functional (billions of devices)
- Linux will not be in low-end, low-power IoT devices
  - e.g. 1M RAM device running on harvested energy
  - I don't expect to see Linux running on a cereal box any time soon
- Core kernel systems are in place to support embedded
  - But new hardware keeps being made
  - We'll always have things to write and upstream
  - Core code can always be improved



CE Workgroup

# Conclusions

- We're not done yet!





CE Workgroup

# Conclusions

- We're not done yet!
- That's OK – we've got job security for the foreseeable future



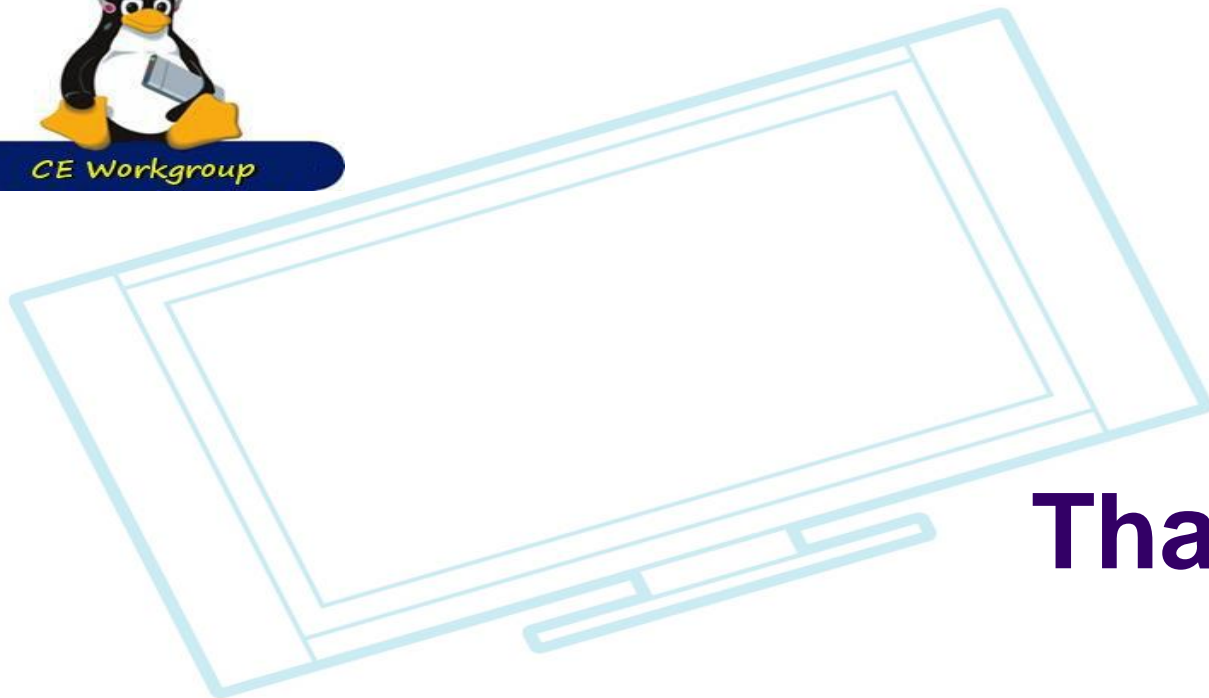


# Conclusions

- We're not done yet!
- That's OK – we've got job security for the foreseeable future
- We can use Linux in our embedded projects!!
- Go forth and continue to use and develop embedded Linux!



CE Workgroup



**Thanks!**

